testfirst

September 15, 2021

[ ]:
```python
# 1. imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[ ]:
```python
# 2. upload the dataset
dataset = pd.read_csv('diabetes.csv')
```

[ ]:
```python
# 3. explore dataset
dataset.describe()
```

[ ]:
```
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000
mean      3.845052  120.894531      69.105469      20.536458   79.799479
std       3.369578   31.972618      19.355807      15.952218  115.244002
min       0.000000    0.000000       0.000000       0.000000    0.000000
25%       1.000000   99.000000      62.000000       0.000000    0.000000
50%       3.000000  117.000000      72.000000      23.000000   30.500000
75%       6.000000  140.250000      80.000000      32.000000  127.250000
max      17.000000  199.000000     122.000000      99.000000  846.000000

              BMI  DiabetesPedigreeFunction         Age     Outcome
count  768.000000                768.000000  768.000000  768.000000
mean    31.992578                  0.471876   33.240885    0.348958
std      7.884160                  0.331329   11.760232    0.476951
min      0.000000                  0.078000   21.000000    0.000000
25%     27.300000                  0.243750   24.000000    0.000000
50%     32.000000                  0.372500   29.000000    0.000000
75%     36.600000                  0.626250   41.000000    1.000000
max     67.100000                  2.420000   81.000000    1.000000
```

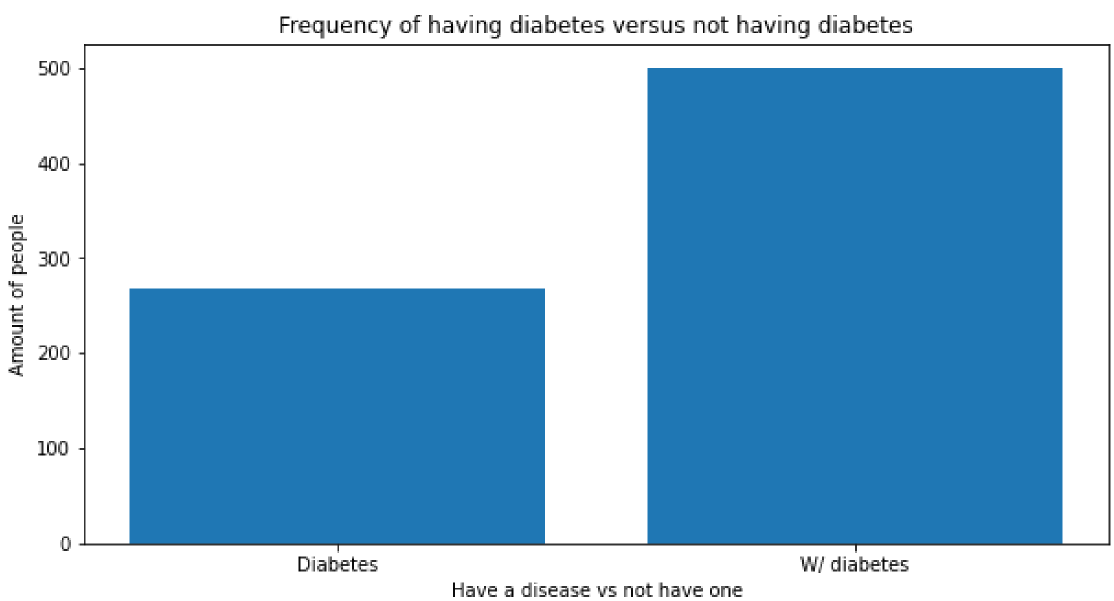
[ ]:
```python
# 4. list variable names
dataset.columns.tolist()
```

[ ]:
```
['Pregnancies',
 'Glucose',
 'BloodPressure',
 'SkinThickness',
```

```
'Insulin',
'BMI',
'DiabetesPedigreeFunction',
'Age',
'Outcome']
```

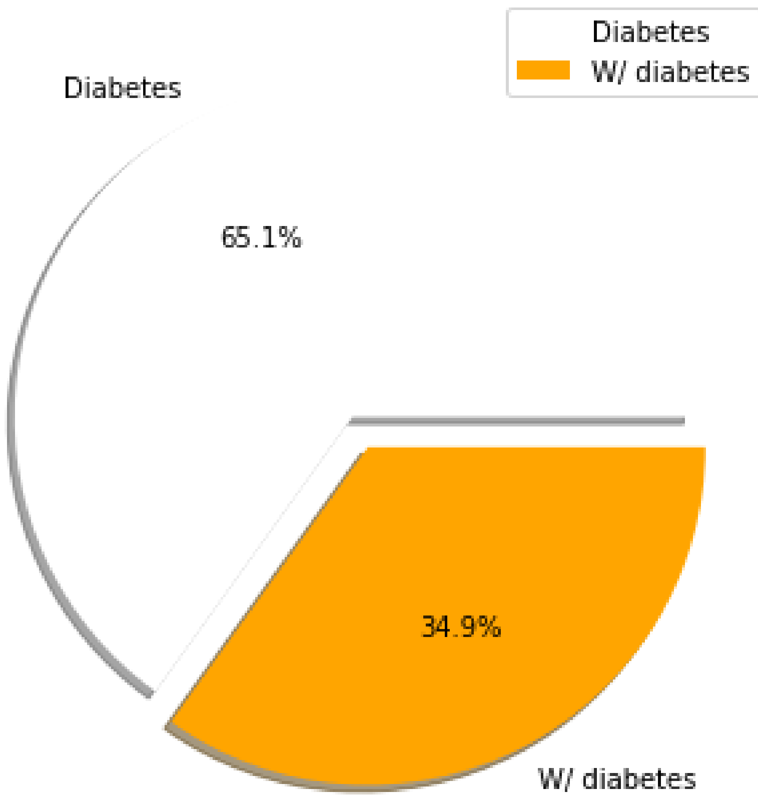
[ ]:
```
# 5. count how many have diabetes and how many don'
value_counts = dataset.Outcome.value_counts()
```

[ ]:
```
# 6. Create a bar chart to represent the frequencies of those with diabetes␣
↪versus those without diabetes,
# label the axes and name the chart.
not_diabetes = value_counts[0]
diabetes = value_counts[1]
diabetes_havers_labels = ['Diabetes', 'W/ diabetes']
plt.bar(diabetes_havers_labels, [diabetes, not_diabetes])
plt.title('Frequency of having diabetes versus not having diabetes')
plt.xlabel('Have a disease vs not have one')
plt.ylabel('Amount of people')
plt.show()
```



[ ]:
```
# 7.        Create a pie plot, change the colors, add the labels, add shade,␣
↪and 'explode' the pie slice for those that have no diabetes.
# Create the legend and put it into the right corner so that it does cover the␣
↪pie plot.
pie_colors = ['white', 'orange']
```

```
plt.pie(value_counts, labels=diabetes_havers_labels, colors=pie_colors,␣
 ↪explode=[0, 0.1], shadow=True, autopct='%1.1f%%')
# matplotlib.rcParams['text.color'] = 'white'
plt.legend()
plt.legend(loc="upper right")
plt.subplots_adjust(left=0.01, bottom=0.001, right=0.75)
plt.show()
```



[ ]:
```
# 8. Calculate and report the mean the standard deviation of the BMI variable.
bmi_mean = dataset.BMI.mean()
bmi_std = dataset.BMI.std()
print('BMI mean: ' + str(bmi_mean))
print('BMI standard deviation: ' + str(bmi_std))
```

```
BMI mean: 31.992578124999998
BMI standard deviation: 7.884160320375446
```

[ ]:
```
# 9. By using the Chebysheffs Theorem infer what is the range for the BMI␣
 ↪variable for 75% of the people in your dataset.
#  What is the range for 89% of the people in your dataset.
```

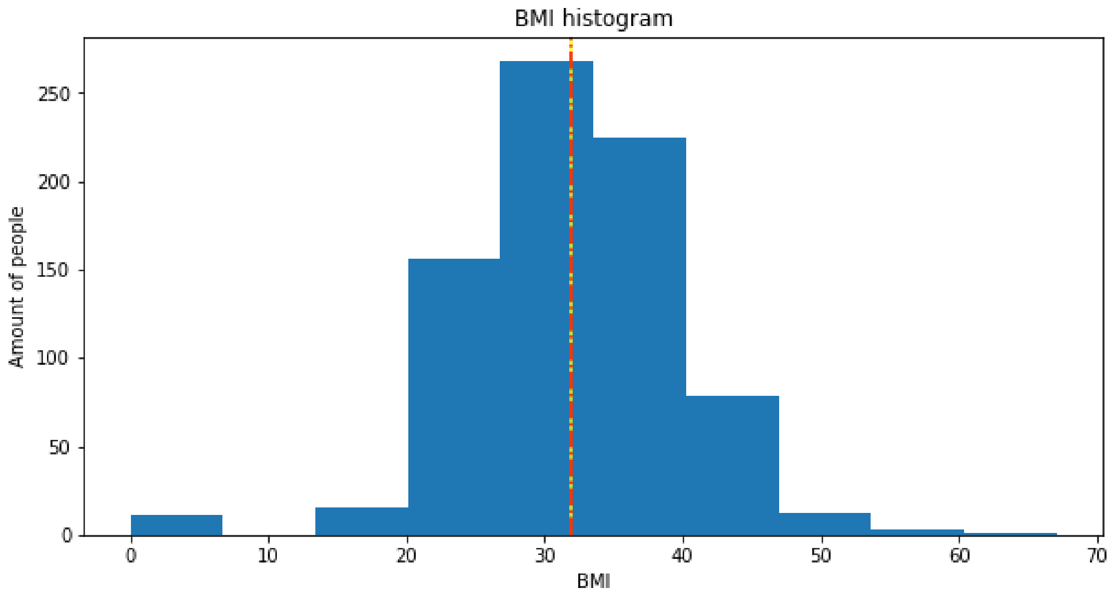
```
# 1) mean - (lower boundary) = within number
# 2) mean - (upper boundary) = within number
#  k - number of standard deviations
#  k = the within number / std
# wn = sqrt(1/(1-x))* std; x = percentage
wn75 = np.sqrt(1/(1 - 75/100)) * bmi_std
wn89 = np.sqrt(1/(1 - 89/100)) * bmi_std
print(wn75)
print('range for 75% is between '+ str(bmi_mean - wn75) + ' and ' +␣
 ↪str(bmi_mean + wn75))
print(wn89)
print('range for 89% is between '+ str(bmi_mean - wn89) + ' and ' +␣
 ↪str(bmi_mean + wn89))
```

```
15.768320640750892
range for 75% is between 16.224257484249108 and 47.76089876575089
23.771637790630525
range for 89% is between 8.220940334369473 and 55.76421591563052
```

[ ]:

```
# 10. Create a histogram for the BMI variable, plot the mean and meadian on the␣
 ↪historgram to answer the following question:
# a)        Is the BMI distribution skewed based on the position of the mean␣
 ↪versus the median?
# b)        Calculate the skewness of BMI
# c)        Based on the value of BMI median and on the chart below infer what␣
 ↪percentage of people in your dataset are obese. (1.5pt)
# create histogram
plt.hist(dataset.BMI)
plt.title('BMI histogram')
plt.xlabel('BMI')
plt.ylabel('Amount of people')
bmi_median = np.median(dataset.BMI)
print('BMI median = '+str(bmi_median))
plt.axvline(bmi_mean, color='yellow', linestyle='-')
plt.axvline(bmi_median, color='red', linestyle='-.')
plt.rcParams['figure.figsize'] = [10, 5]
plt.show()
# a: it doesn't seem that much skewed based on any of these  variables; on the␣
 ↪other hand, mean is not in the middle of highest bar
# b: skewness is negative, therefore there are tails on the left side
bmi_skewness = dataset.BMI.skew()
print('BMI skewness = '+str(bmi_skewness))
# c:
# obese people are with bmi = 30 or higher
print('there is this exact percantage of people with obesity: '+␣
 ↪str(len(dataset.BMI[dataset.BMI >= 30])/(len(dataset.BMI) / 100)))
```

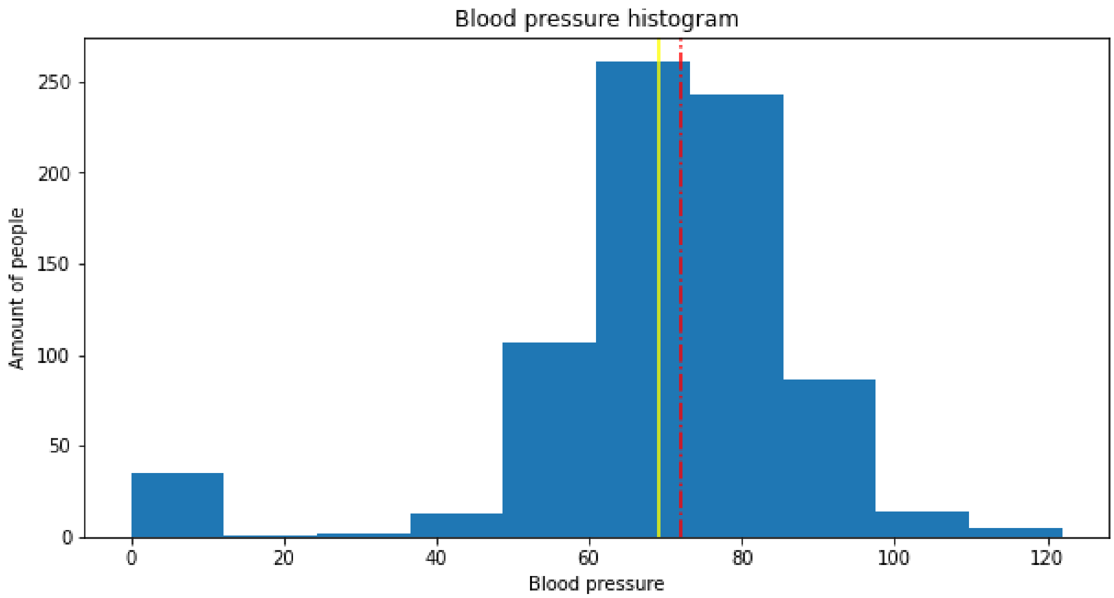
```
BMI median = 32.0
```




```
BMI skewness = -0.42898158845356543
there is this exact percantage of people with obesity: 61.458333333333336
```

[ ]:
```python
# 11. doing step 10 for blood pressure
plt.hist(dataset.BloodPressure)
plt.title('Blood pressure histogram')
plt.xlabel('Blood pressure')
plt.ylabel('Amount of people')
blood_pressure_median = np.median(dataset.BloodPressure)
blood_pressure_mean = np.mean(dataset.BloodPressure)
print('Blood pressure median = '+str(blood_pressure_median))
plt.axvline(blood_pressure_mean, color='yellow', linestyle='-')
plt.axvline(blood_pressure_median, color='red', linestyle='-.')
plt.rcParams['figure.figsize'] = [10, 5]
plt.show()
# a: distribution is skewed  on the mean more it seems, but i can completely␣
↪misunderstand the theory at this point
# b: calculate skewness, it i negative, therefore tails are on the left, but it␣
↪is also VERY big, so we should stick to median instead of mean
blood_pressure_skewness = dataset.BloodPressure.skew()
print('Blood pressure skewness = '+str(blood_pressure_skewness))
# c:
# obesity is not the thing here
# I found out that with more outliers on the left the mean value became less␣
↪descriptive and according tothe studies and to te skewness we should
# stick to the median instead
```

```
Blood pressure median = 72.0
```



Blood pressure skewness = -1.8436079833551302

[ ]:
```
# 12. Remove all the observations where the BloodPressure' and the BMI␣
↪variables take the value of zero
dataset = dataset.drop(dataset[dataset.BloodPressure == 0].index)
dataset = dataset.drop(dataset[dataset.BMI == 0].index)
dataset.describe()
```

[ ]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin
count	729.000000	729.000000	729.000000	729.000000	729.000000
mean	3.858711	121.046639	72.367627	21.499314	83.946502
std	3.357468	32.255215	12.375838	15.708376	116.803000
min	0.000000	0.000000	24.000000	0.000000	0.000000
25%	1.000000	99.000000	64.000000	0.000000	0.000000
50%	3.000000	117.000000	72.000000	24.000000	46.000000
75%	6.000000	141.000000	80.000000	33.000000	130.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000

	BMI	DiabetesPedigreeFunction	Age	Outcome
count	729.000000	729.000000	729.000000	729.000000
mean	32.469959	0.474117	33.318244	0.344307
std	6.885098	0.331649	11.753078	0.475468
min	18.200000	0.078000	21.000000	0.000000
25%	27.500000	0.245000	24.000000	0.000000
50%	32.400000	0.378000	29.000000	0.000000
75%	36.600000	0.627000	41.000000	1.000000
max	67.100000	2.420000	81.000000	1.000000

[ ]:
```
# 13.8 Now perform once again all the steps in 8, 9, 10 and 11
# 8. Calculate and report the mean the standard deviation of the BMI variable.
bmi_mean = dataset.BMI.mean()
bmi_std = dataset.BMI.std()
print('BMI mean: ' + str(bmi_mean))
print('BMI standard deviation: ' + str(bmi_std))
# mean is now bigger, std is now less
```

```
BMI mean: 32.46995884773663
BMI standard deviation: 6.885098188024897
```

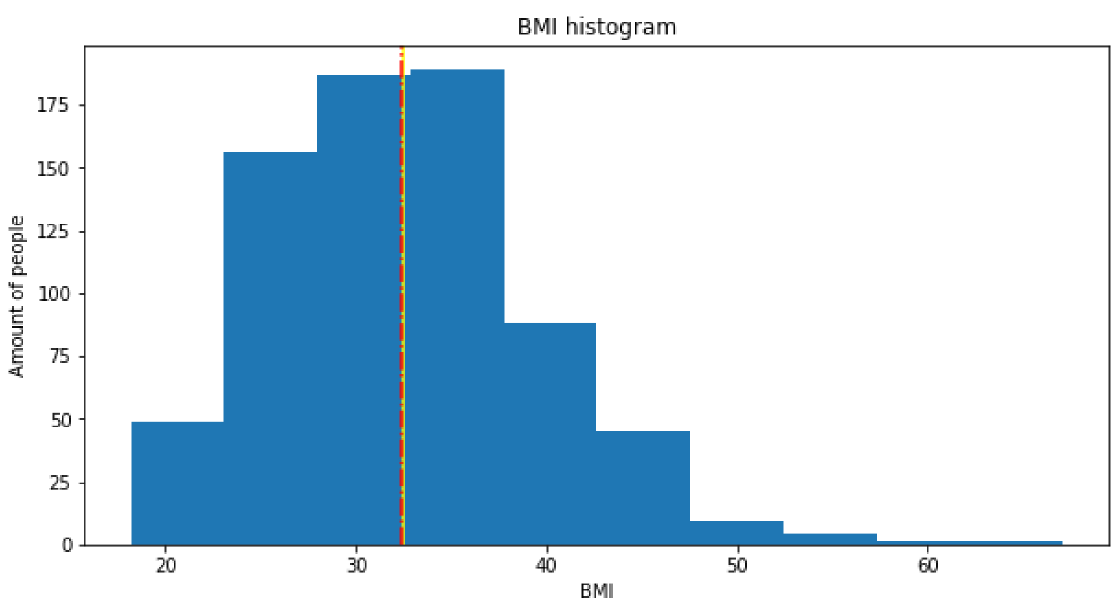
[ ]:
```
# 13.9. By using the Chebysheffs Theorem infer what is the range for the BMI␣
 ↪variable for 75% of the people in your dataset.
#  What is the range for 89% of the people in your dataset.
# 1) mean - (lower boundary) = within number
# 2) mean - (upper boundary) = within number
#  k - number of standard deviations
#  k = the within number / std
# wn = sqrt(1/(1-x))* std; x = percentage
wn75 = np.sqrt(1/(1 - 75/100)) * bmi_std
wn89 = np.sqrt(1/(1 - 89/100)) * bmi_std
print(wn75)
print('range for 75% is between '+ str(bmi_mean - wn75) + ' and ' +␣
 ↪str(bmi_mean + wn75))
print(wn89)
print('range for 89% is between '+ str(bmi_mean - wn89) + ' and ' +␣
 ↪str(bmi_mean + wn89))
# as a result within numbers are smaller, the range also became smaller because␣
 ↪zeros seemed to be just outliers that were spoiling the results
```

```
13.770196376049794
range for 75% is between 18.69976247168683 and 46.240155223786424
20.75935212221311
range for 89% is between 11.710606725523519 and 53.229310969949736
```

[ ]:
```
# 13.10. Create a histogram for the BMI variable, plot the mean and meadian on␣
 ↪the historgram to answer the following question:
# a)        Is the BMI distribution skewed based on the position of the mean␣
 ↪versus the median?
# b)        Calculate the skewness of BMI
# c)        Based on the value of BMI median and on the chart below infer what␣
 ↪percentage of people in your dataset are obese. (1.5pt)
# create histogram
plt.hist(dataset.BMI)
plt.title('BMI histogram')
plt.xlabel('BMI')
plt.ylabel('Amount of people')
bmi_median = np.median(dataset.BMI)
```

```
print('BMI median = '+str(bmi_median))
plt.axvline(bmi_mean, color='yellow', linestyle='-')
plt.axvline(bmi_median, color='red', linestyle='-.')
plt.rcParams['figure.figsize'] = [10, 5]
plt.show()
bmi_skewness = dataset.BMI.skew()
print('BMI skewness = '+str(bmi_skewness))
# c:
# obese people are with bmi = 30 or higher
print('there is this exact percantage of people with obesity: '+␣
↪str(len(dataset.BMI[dataset.BMI >= 30])/(len(dataset.BMI) / 100)))
# skewness somehow is now positive and bigger, obesity without those zeroes␣
↪also appears more often
# mean and median differ more Now
# how can skewness be bigger when i removed the outliers?..
# also they are no longer on the highest bar, which is also weird for median i␣
↪guess
```

```
BMI median = 32.4
```

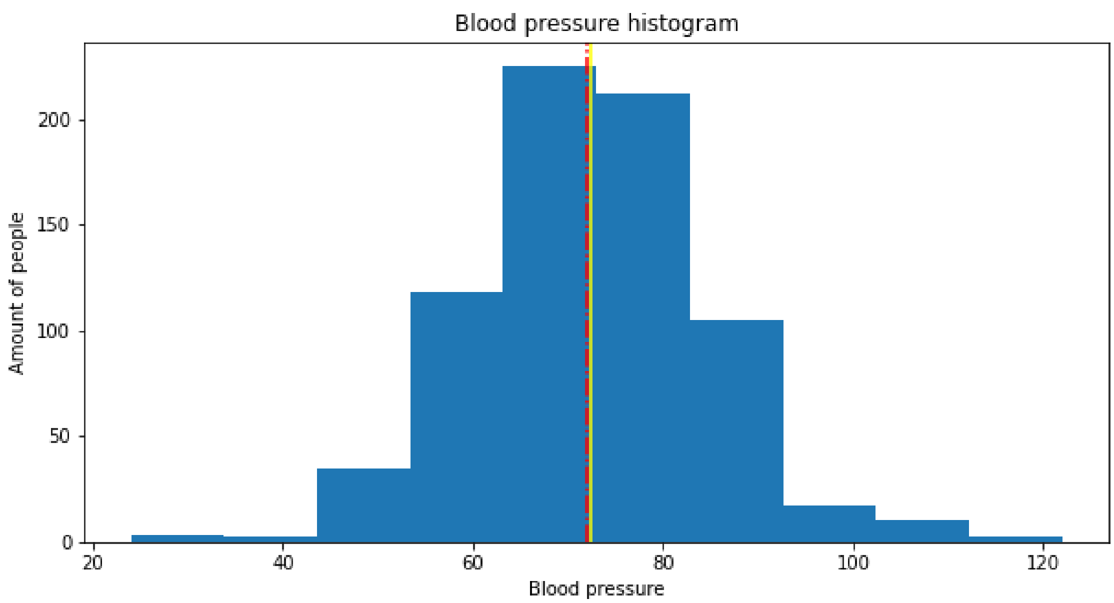



```
BMI skewness = 0.5951875009633939
there is this exact percantage of people with obesity: 62.55144032921811
```

[ ]:
```
# 13.11. doing step 10 for blood pressure
plt.hist(dataset.BloodPressure)
plt.title('Blood pressure histogram')
plt.xlabel('Blood pressure')
plt.ylabel('Amount of people')
```

```
blood_pressure_median = np.median(dataset.BloodPressure)
blood_pressure_mean = np.mean(dataset.BloodPressure)
print('Blood pressure median = '+str(blood_pressure_median))
plt.axvline(blood_pressure_mean, color='yellow', linestyle='-')
plt.axvline(blood_pressure_median, color='red', linestyle='-.')
plt.rcParams['figure.figsize'] = [10, 5]
plt.show()
# a: distribution is skewed  on the mean more
# b: calculate skewness, it is positive and also very small, therefore tails␣
 ↪are on the right, so we should stick to mean
blood_pressure_skewness = dataset.BloodPressure.skew()
print('Blood pressure skewness = '+str(blood_pressure_skewness))
# c: obesity is not the thing here
```

```
Blood pressure median = 72.0
```




```
Blood pressure skewness = 0.13445955512111737
```